# STAT 672 HW 5

Tom Wallace

April 13, 2018

## I-D

10-fold cross validation error is 0.19

This error rate is lower than we would expect based on the data-generating process described in **1A**. There is no true relationship between $\boldsymbol{X}_i$ and $Y_i$ and so in theory the logistic regression model should be no more effective than random guessing. We thus would expect a cross-validation error of about 0.5. The better-than-expected performance is attributable to two factors.

One, $D >> n$. Simply by random chance some features will be highly correlated with the label. Our regularization process selected these features to be part of the model. If $D$ is not so large relative to $n$, model performance drops to the expected level. For example, I ran the simulation with $n = 10000$ and $D = 100$, and obtained a cross-validation error of 0.49.

Two, we violated the central premise of cross-validation—a strict segregation between training and validation sets—by conducting some model training on the full data-set. The regularization process in **1B** selected a subset of features $\boldsymbol{X_S}$ with a particularly strong relationship with label $\boldsymbol{Y}$; we only conducted cross-validation *after* this feature selection was performed on the *whole* dataset. In this sense we "cheated" and allowed the training process to incorporate information from the validation set. A correct implementation of cross-validation would conduct the regularization process of **1B** only on the training set, and doing so would cause the better-than-expected performance to disappear.

## II-A

The best models and the associated cross-validation error is given below. The numbers in the left-hand column are the indices of features included in the model.

| **Features in Model** | **5-Fold CV Error** |
|---|---|
| 0 3 4 5 6 9 11 14 | 0.225 |
| 0 3 4 5 6 9 11 15 | 0.225 |
| 1 3 4 5 6 8 9 13 14 | 0.225 |
| 1 3 4 5 6 8 9 13 | 0.225 |
| 1 3 8 9 10 11 13 14 | 0.225 |
| 1 3 8 9 10 11 13 | 0.225 |
| 2 3 9 10 14 | 0.225 |
| 2 3 9 10 | 0.225 |
| 3 4 5 6 9 13 14 | 0.225 |
| 3 4 5 6 9 13 | 0.225 |
| 3 6 7 8 9 10 14 | 0.225 |
| 3 6 7 8 9 10 | 0.225 |

## II-B

The final classifier[1] trained on the whole training data attains an error rate of 0.502 against the testing data (`Xtest, Ytest`). This is much higher than the cross-validation error of 0.225 obtained in **II-B**, and in fact is about what one would get from random guessing. There are several reasons for this underperformance.

One, $n_{\text{train}} << n_{\text{test}}$. Because the model is trained on such a small dataset, it did not learn the general relationship between features and label. Note that the mathematical proof of why validation works (slides 28-29 of class 6) implies that the probability of producing the optimal model increases with the size of the validation set. It thus is unsurprising that our small validation set failed to produce a good model.

Two, our classifier lacks a regularization parameter, and so it overfits to the training data. This results in achieving low empirical error (i.e. good performance on the training set) but high generalization error (i.e. poor performance on the training set).

## II-C

Randomly permuting the labels in `Ytrain` could expose the aforementioned issues without the need for a test set. Randomly permuting the labels would mean there is no true relationship between feature and labels, and so a model attempting to predict label based on features should be no better than random guessing (i.e. should achieve cross-validation error of about 0.5). If the model achieves much lower error than this, there is something wrong with the feature selection and/or classifier.

## II-D

[1] As shown in the preceding table, there are multiple models that achieve the lowest cross-validation error, so really this sentence should be final classifier*s*. I tried several, and all led to the same phenomenon.